

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-330794

(43)Date of publication of application : 30.11.2000

(51)Int.Cl.

G06F 9/45

(21)Application number : 11-169869

(71)Applicant : NEC CORP

(22)Date of filing : 16.06.1999

(72)Inventor : HATAKEYAMA AKEMI

(30)Priority

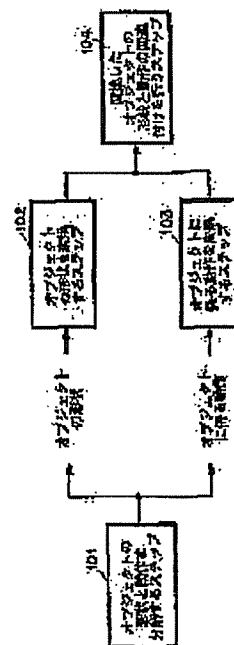
Priority number : 11070434 Priority date : 16.03.1999 Priority country : JP

(54) METHOD FOR CONVERTING MHEG DOCUMENT INTO HTML DOCUMENT

(57)Abstract:

PROBLEM TO BE SOLVED: To convert an MHEG object which can not be converted as a component by converting an MHEG document into an HTML document which has a description of an object, and a description and a script of CSS.

SOLUTION: The shape and operation of a slider object and a link object are extracted from a document described in MHEG and decomposed. The shape of the slider object is converted into the properties of multiple objects described in HTML. Other properties of the slider object are converted into variables described by a script of HTML. Link effect regarding the slider object described in the link object is converted into a method described by a script of HTML. The event type of the link object is converted into the event handler of an object of HTML. The shape and operation represented in the generated HTML are related.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-330794
(P2000-330794A)

(43) 公開日 平成12年11月30日 (2000. 11. 30)

(51) Int. Cl.
G 0 6 F 9/45

識別記号

F I
G 0 6 F 9/44

テーマコード(参考)
3 2 2 Z 5 B 0 8 1

審査請求 有 請求項の数20 OL (全 27 頁)

(21) 出願番号 特願平11-169869

(22) 出願日 平成11年6月16日 (1999. 6. 16)

(31) 優先権主張番号 特願平11-70434

(32) 優先日 平成11年3月16日 (1999. 3. 16)

(33) 優先権主張国 日本 (J P)

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 島山 朱美

東京都港区芝五丁目7番1号 日本電気株式会社内

(74) 代理人 100065385

弁理士 山下 稔平

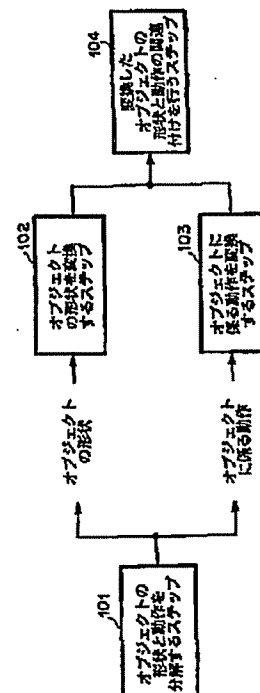
Fターム(参考) 5B081 AA10 BB08

(54) 【発明の名称】 MHEG文書をHTML文書に変換する方法

(57) 【要約】

【課題】 部品として変換できないMHEGオブジェクトを変換できるMHEG文書からHTML文書への変換方法を提供する。

【解決手段】 MHEG文書中の第1 MHEGオブジェクトのプロパティをHTML文書上の1以上のオブジェクトのプロパティに変換し、第1 MHEGオブジェクトのHTML文書上のオブジェクトのプロパティに変換されるプロパティ以外のプロパティをHTML文書上のスクリプトで記述される変数に変換し、リンクオブジェクトのイベントタイプをリンクオブジェクトのイベントソースに対応したHTML文書上の第2オブジェクトに関連付けられたイベントハンドラに変換し、第2 MHEGオブジェクトをイベントソースとして持つリンクオブジェクトのリンクイベントをイベントハンドラに関連づけられ且つスクリプトにより記述されるメソッドに変換する。



【特許請求の範囲】

【請求項1】 1又は2以上のMHEG (Multimedia and Hypermedia Expert Group)オブジェクトを有するMHEG文書中を、1又は2以上のオブジェクトの記述、CSS (Cascading Style Sheet)の記述及びスクリプトを有するHTML (Hypertext Mark-up Language)文書に変換することを特徴とするMHEG文書をHTML文書に変換する方法。

【請求項2】 前記HTML文書のオブジェクトにはプラグインオブジェクトが含まれることを特徴とする請求項1に記載のMHEG文書をHTML文書に変換する方法。

【請求項3】 MHEG (Multimedia and Hypermedia Expert Group)文書中の第1のMHEGオブジェクトのプロパティの少なくとも一部をHTML (Hypertext Mark-up Language)文書上の1又は2以上のオブジェクトのプロパティに変換するステップと、

前記第1のMHEGオブジェクトの前記HTML文書上のオブジェクトのプロパティに変換されるプロパティ以外のプロパティの少なくとも一部をHTML文書上のスクリプトにより記述される変数に変換するステップと、第2のMHEGオブジェクトをイベントソースとして持つリンクオブジェクトのイベントタイプを該リンクオブジェクトの前記イベントソースに対応したHTML文書上のオブジェクトに関連付けられたイベントハンドラに変換するステップと、

前記第2のMHEGオブジェクトをイベントソースとして持つ前記リンクオブジェクトのリンクイベントを前記イベントハンドラに関連づけられ且つ前記スクリプトにより記述されるメソッド又はメソッドの組み合わせに変換するステップと、

を有し、前記第1のMHEGオブジェクトと前記第2のMHEGオブジェクトは、同一であるか又は異なることを特徴とするMHEG文書をHTML文書に変換する方法。

【請求項4】 前記HTML文書上のオブジェクトのプロパティに変換されるプロパティは前記第1のMHEGオブジェクトの形状プロパティの少なくとも一部を含むことを特徴とする請求項3に記載のMHEG文書をHTML文書に変換する方法。

【請求項5】 前記メソッド又はメソッドの組み合わせは、前記第1のMHEGオブジェクトのプロパティより変換された前記HTML文書上のオブジェクトのプロパティを扱うことを特徴とする請求項3又は4に記載のMHEG文書をHTML文書に変換する方法。

【請求項6】 前記メソッド又はメソッドの組み合わせは、前記第1のMHEGオブジェクトのプロパティより変換された前記HTML文書上のオブジェクトのプロパティを取得することを特徴とする請求項5に記載のMHEG文書をHTML文書に変換する方法。

【請求項7】 前記メソッド又はメソッドの組み合わせは、前記第1のMHEGオブジェクトのプロパティより変換された前記HTML文書上のオブジェクトのプロパティを変化させることを特徴とする請求項5に記載のMHEG文書をHTML文書に変換する方法。

【請求項8】 前記第1のMHEGオブジェクトのプロパティより変換されるプロパティを有するHTML文書上のオブジェクトは、一対のDIVタグ及び該一対のDIVタグに挟まれるスタイル属性の記述文であることを特徴とする請求項3乃至7のいずれか1項に記載のMHEG文書をHTML文書に変換する方法。

【請求項9】 前記スクリプト内で同一種類の複数のMHEGオブジェクトより変換されたHTML文書上のオブジェクトを配列で管理することを特徴とする請求項3乃至8のいずれか1項に記載のMHEG文書をHTML文書に変換する方法。

【請求項10】 前記スクリプトはジャバスクリプト (JavaScript)であることを特徴とする請求項3乃至9のいずれか1項に記載のMHEG文書をHTML文書に変換する方法。

【請求項11】 前記第1のMHEGオブジェクトはスライダオブジェクトであることを特徴とする請求項3乃至10のいずれか1項に記載のMHEG文書をHTML文書に変換する方法。

【請求項12】 前記第1及び第2のMHEGオブジェクトはパレットオブジェクトではなく、前記MHEG文書中の前記パレットオブジェクトをカラーのスタイル属性に変換するステップを有することを特徴とする請求項3乃至11のいずれか1項に記載のMHEG文書をHTML文書に変換する方法。

【請求項13】 前記第1及び第2のMHEGオブジェクトは変数オブジェクトではなく、MHEG文書中の前記変数オブジェクトをスクリプトの変数に変換するステップを有することを特徴とする請求項3乃至11のいずれか1項に記載のMHEG文書をHTML文書に変換する方法。

【請求項14】 MHEG文書中の前記第1又は第2のオブジェクト又はその他のMHEGオブジェクトの少なくとも一部をHTML文書上のプラグインオブジェクトに変換するステップを有することを特徴とする請求項3乃至11のいずれか1項に記載のMHEG文書をHTML文書に変換する方法。

【請求項15】 MHEG文書中の前記第1又は第2のオブジェクト又はその他のMHEGオブジェクトの少なくとも一部をHTML文書上のスクリプトに変換するステップを有することを特徴とする請求項1乃至11のいずれか1項に記載のMHEG文書をHTML文書に変換する方法。

【請求項16】 MHEG文書中のパレットオブジェクトをカラーのスタイル属性に変換するステップを有する

ことを特徴とするMHEG文書をHTML文書に変換する方法。

【請求項17】 MHEG文書中の変数オブジェクトをスクリプトの変数に変換するステップを有することを特徴とするMHEG文書をHTML文書に変換する方法。

【請求項18】 MHEG文書中のMHEGオブジェクトの少なくとも一部をHTML文書上のプラグインオブジェクトに変換するステップを有することを特徴とするMHEG文書をHTML文書に変換する方法。

【請求項19】 MHEG文書中のMHEGオブジェクトの少なくとも一部をHTML文書上のスクリプトに変換するステップを有することを特徴とするMHEG文書をHTML文書に変換する方法。

【請求項20】 請求項1乃至19のいずれか1項に記載のMHEG文書をHTML文書に変換する方法をコンピュータに実行させるためのプログラムを記録したことを特徴とするコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、MHEG (Multimedia and Hypermedia Expert Group) 文書をHTML (Hypertext Mark-up Language) 文書に変換する方法に関し、特に、HTML文書上のオブジェクトに直接変換することができないMHEGオブジェクトを含むMHEG文書をHTML文書に変換する方法に関する。

【0002】

【従来の技術】デジタル放送の開始に合わせて、データ放送も順次始まりつつある。データ放送の符号化方式として、或いは、セットトップユニットのハイレベルのAPI (Application Programming Interface) の標準として、DAVIC (Digital Audio-Visual Council) がMHEG-5を採用した。MHEGは、マルチメディア情報のオブジェクトやハイパーメディア情報のオブジェクトの表現 (representation) を交換 (interchange) するための標準であり、ISO/IEC (International Organization for Standardization/International Electrotechnical Commission) 及びITU (International Telecommunication Union) の標準である。MHEGのモデルはクラス (「Ingredient Class」という。) より生成されるインスタンスとしてのオブジェクト (MHEGオブジェクト、「Ingredient」という。) として記述される。MHEG文書は、原始的にはISOの規格であるASN1 (Abstract Syntax Notation version 1) により記述され、バイナリコードに変換される。

【0003】一方で、インターネットでは、ホームページ記述言語としてはHTML 4.0が標準となってきた。

【0004】これまで、MHEG文書とHTML文書は別々に使用されることが想定されていたが、WWW (World Wide Web) の表現能力も高まり、それぞれのコンテンツを相互利用するニーズが高まってきている。

【0005】

【発明が解決しようとする課題】しかしながら、MHEG-5とHTML 4.0、CSS (Cascading Style Sheet)、JavaScript (ジャバスクリプト) の組み合わせとを比較すると機能的には似ているが、部品として変換できないMHEGオブジェクトがあり、今後のインターネットと放送のコンテンツ相互利用を考えるとそのような機能の変換が必須である。

【0006】そこで、本発明は、部品として変換することができないMHEGオブジェクトを変換することができるMHEG文書をHTML文書に変換する方法を提供することを目的とする。

【0007】また、本発明は、部品として変換することができないスライダオブジェクトを変換することができるMHEG文書をHTML文書に変換する方法を提供することを目的とする。

【0008】

【課題を解決するための手段】本発明によるMHEG文書をHTML文書に変換する方法は、1又は2以上のMHEGオブジェクトを有するMHEG文書中を、1又は2以上のオブジェクトの記述、CSSの記述及びスクリプトを有するHTML文書に変換することを特徴とする。

【0009】また、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記HTML文書のオブジェクトにはプラグインオブジェクトが含まれることを特徴とする。更に、本発明によるMHEG文書をHTML文書に変換する方法は、MHEG文書中の第1のMHEGオブジェクトのプロパティの少なくとも一部をHTML文書上の1又は2以上のオブジェクトのプロパティに変換するステップと、前記第1のMHEGオブジェクトの前記HTML文書上のオブジェクトのプロパティに変換されるプロパティ以外のプロパティの少なくとも一部をHTML文書上のスクリプトにより記述される変数に変換するステップと、第2のMHEGオブジェクトをイベントソースとして持つリンクオブジェクトのイベントタイプを該リンクオブジェクトの前記イベントソースに対応したHTML文書上のオブジェクトに関連付けられたイベントハンドラに変換するステップと、前記第2のMHEGオブジェクトをイベントソースとして持つ前記リンクオブジェクトのリンクイベントを前記イベントハンドラに関連づけられ且つ前記スクリプトにより記述されるメソッド又はメソッドの組み合わせに変換するステップと、を有し、前記第1のMHEGオブジェクトと前記第2のMHEGオブジェクトは、同一であるか又は異なることを特徴とする。

【0010】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記HTML文書上の

オブジェクトのプロパティに変換されるプロパティは前記第1のMHEGオブジェクトの形状プロパティの少なくとも一部を含むことを特徴とする。

【0011】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記メソッド又はメソッドの組み合わせは、前記第1のMHEGオブジェクトのプロパティより変換された前記HTML文書上のオブジェクトのプロパティを扱うことを特徴とする。

【0012】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記メソッド又はメソッドの組み合わせは、前記第1のMHEGオブジェクトのプロパティより変換された前記HTML文書上のオブジェクトのプロパティを取得することを特徴とする。

【0013】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記メソッド又はメソッドの組み合わせは、前記第1のMHEGオブジェクトのプロパティより変換された前記HTML文書上のオブジェクトのプロパティを変化させることを特徴とする。

【0014】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記第1のMHEGオブジェクトのプロパティより変換されるプロパティを有するHTML文書上のオブジェクトは、一対のDIVタグ及び該一対のDIVタグに挟まれるスタイル属性の記述文であることを特徴とする。

【0015】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記スクリプト内で同一種類の複数のMHEGオブジェクトより変換されたHTML文書上のオブジェクトを配列で管理することを特徴とする。

【0016】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記スクリプトはジャバスクリプト(JavaScript)であることを特徴とする。

【0017】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記第1のMHEGオブジェクトはスライドオブジェクトであることを特徴とする。

【0018】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記第1及び第2のMHEGオブジェクトはパレットオブジェクトではなく、前記MHEG文書中の前記パレットオブジェクトをカラーのスタイル属性に変換するステップを有することを特

徴とする。

【0019】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記第1及び第2のMHEGオブジェクトは変数オブジェクトではなく、MHEG文書中の前記変数オブジェクトをスクリプトの変数に変換するステップを有することを特徴とする。

【0020】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、MHEG文書中の前記第1又は第2のオブジェクト又はその他のMHEGオブジェクトの少なくとも一部をHTML文書上のプラグインオブジェクトに変換するステップを有することを特徴とする。更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、MHEG文書中の前記第1又は第2のオブジェクト又はその他のMHEGオブジェクトの少なくとも一部をHTML文書上のスクリプトに変換するステップを有することを特徴とする。

【0021】更に、本発明によるMHEG文書をHTML文書に変換する方法は、MHEG文書中のパレットオブジェクトをカラーのスタイル属性に変換するステップを有することを特徴とする。

【0022】更に、本発明によるMHEG文書をHTML文書に変換する方法は、MHEG文書中の変数オブジェクトをスクリプトの変数に変換するステップを有することを特徴とする。

【0023】更に、本発明によるMHEG文書をHTML文書に変換する方法は、MHEG文書中のMHEGオブジェクトの少なくとも一部をHTML文書上のプラグインオブジェクトに変換するステップを有することを特徴とする。

【0024】更に、本発明によるMHEG文書をHTML文書に変換する方法は、MHEG文書中のMHEGオブジェクトの少なくとも一部をHTML文書上のスクリプトに変換するステップを有することを特徴とする。

【0025】本発明によるコンピュータ読み取り可能な記録媒体は、上記のMHEG文書をHTML文書に変換する方法をコンピュータに実行させるためのプログラムを記録したことを特徴とする。

【0026】

【発明の実施の形態】
【実施形態1】本発明の実施形態1においては、特にスライドオブジェクトの変換方法を例に取り説明する。

【0027】本実施形態によるスライドオブジェクトの変換方法は、スライドオブジェクトを持つマルチメディア記述言語であるMHEGで記述された文書を、スライドオブジェクトを持たない他のマルチメディア記述言語であるHTMLで記述された文書に変換する際の、スライドオブジェクトの変換方法である。本実施形態は、変

換元の言語により記述されるスライダオブジェクトおよびその機能を、変換先の言語により記述される他のオブジェクトおよび機能の組み合わせにより実現し、変換先の言語にスライダオブジェクトがなくても動作に支障のないよう変換できることを特徴とする。

【0028】本発明の実施形態として、MHEG-5により記述されたスライダオブジェクトからHTML 4.0, CSS, JavaScriptの組み合わせ（以下、「HTML 4.0 set」という。）への変換を示す。

【0029】以下、本発明の構成を図面を参照して説明する。

【0030】図1は、本発明の実施形態によるスライダオブジェクトの変換方法を示すブロック図である。図1において、101はMHEG-5で記述されたアプリケーションおよびシーンの中からスライダオブジェクトの形状と動作、及びスライダオブジェクトへのアクションを持っているLinkオブジェクト（リンクオブジェクト）の抽出と分解を行うステップである。102はスライダオブジェクトのプロパティの一部である形状を変換するステップであり、スライダオブジェクトの形状をHTML 4.0 setにより記述される複数のオブジェクトのプロパティに変換する。また、102では、スライダオブジェクトの他のプロパティを、HTML 4.0 set上のJavaScriptにより記述される変数に変換する。103はLinkオブジェクト内に記述されたスライダオブジェクトに係るリンクオブジェクトをHTML 4.0 set上のJavaScriptにより記述されるメソッド又はメソッドの組み合わせに変換するステップである。また、このステップで、Linkオブジェクトのイベントタイプは当該Linkオブジェクトのイベントソースに対応するHTML 4.0 set上のオブジェクトのイベントハンドラに変換される。104は、102および103により生成されたHTML 4.0 setで表現された形状および動作の関連付けを行うステップである。このステップで、メソッド又はメソッドの組み合わせは、リンクオブジェクトのイベントタイプに対応するHTML 4.0 set上のイベントハンドラを扱うこととなる。

【0031】次に、本発明の実施形態の動作について説明する。

【0032】本実施形態ではMHEG-5の1アプリケーションおよび1シーンをHTML 4.0 setの1ファイル（ページ）に変換することを想定している。

【0033】本発明の実施形態のうち、スライダオブジェクトの変換については、図2に示すようにスライダをインジケータと枠の2つの部品の組み合わせとして変換する。具体的には各部品を矩形として考え、DIVタグであらわす。また、MHEG-5記述における属性値からHTML 4.0 setにおける属性値への変換は図3に示すルールにもとづいて行う。図3の属性欄にはMHEG-5のスライダオブジェクトが持ちうる全ての属性値を示した。種別の欄がM（マンドトリ（Mandatory）の略。）である属性は、必

須の属性であり、O（オプション(option)を意味する。）である属性は、記述しなくても良い属性である。Oに続くかっこ内の内容は、デフォルト値である。HTML 4.0 set表現の欄は、MHEG-5の各属性がHTML 4.0 setではどのような記述に変換されるかを示している。表現はプロパティである。このプロパティは、styleが先頭に付き、CSSで用いられるスタイル属性である。また、備考欄にはそれらがどのような役割を担うかを示した。HTML 4.0 setの表現欄において“-”は変換には使用しないことを示し、“※”は直接的な変換ができないことを示す。

【0034】以下に各属性値の変換の詳細について述べる。

【0035】Sharedは、MHEG-5アプリケーションの全てのシーンにおいて、このオブジェクトを共通使用するか否かを示す属性である。もし、Sharedがtrueであるならば、各シーンに対応するすべてのHTML 4.0 setファイル中にこのスライダオブジェクトに対応する2つのDIVタグを記述する。OriginalPaletteRefは、カラー属性(HighlightRefColor, スライダーrefColor)に対するカラー指定をパレット参照により行う場合に、表記しなければならない属性である。もし、パレットを使わず直接的に、たとえばRGBカラーで指定するのであれば、必要ない属性である。Orientation属性は、スライダの向き、方向を示す属性である。取りうる値としてup/down/left/rightがあり、MinValueからMaxValueに向かう方向を示している。

【0036】ObjectIdentifierはIDに変換される。仮に、ObjectIdentifier=1000だった場合、スライド部分をID=1000s、外枠部分をID=1000oのように変換する。

【0037】InitiallyActiveはデフォルトの可視性を示す。trueならばstyle.visibility="visible", falseならばstyle.visibility="hidden"とする。

【0038】OriginalBoxSizeは外枠部分の大きさを示し、OriginalPositionは左上の位置を示すので、それぞれstyle.width, style.heightとstyle.left, style.topに変換する。

【0039】HighlightRefColor, SliderRefColorはそれぞれハイライト色、スライダー色を示し、パレット参照で指定する場合と、直接値が入る場合がある。パレット指定している場合は、直接的な指定に変換し、HTML 4.0 setでは*.color=#00FFFFのような指定に変換する。

【0040】MinValue, MaxValueはOrigBoxSizeで示された範囲の両端を示す。このとき、Orientationで指定されている方向指定にのっとり、MinとMaxに対応する端を定める。

【0041】InitialValueは初期状態でのスライダの相対的な位置であり、MinValue ≤ InitialValue ≤ MaxValueの関係にある。

【0042】SliderStyleはNormal/Thermometer/Proportionalの3種類をとる。各スタイルの概観例を、Normalは図4(a)に、Thermometerは図4(b)に、Proportionalは図4(c)に示した。

【0043】InitialPortionはOrientation = proportionalの場合のみに有効な属性であり、スライド部分の大きさ(幅)を指定する。

【0044】これらの属性を用いて図5(a)に示すスライドオブジェクトを、HTML 4.0 setに変換した結果は、図5(b)のようになる。

【0045】次に、スライドオブジェクトに係るリンクイフェクトの要素であるアクションの変換方法を示す。図6には各アクションのMHEG-5での記述形式、HTML 4.0 set(JavaScript記述)に変換した場合の関数の引数概要、各アクションの動作を示した。それぞれのJavaScriptで記述された関数の詳細を図7の(x)、(a)と図8の(b)~(e)に示す。なお、各関数の中で使用されている変数のうち、StepSizeとOrientationとSliderTypeはページ読み込み時にJavaScriptにより記述される変数として図7(x)の様に設定する。

【0046】以下に関数の詳細説明を示す。

【0047】図7(x)のInitialize()で定義される変数はMHEG-5のスライドの属性値であって、HTML 4.0のstyle指定では示せない属性に対応する。図7(a)のStep(IDs, ステップ数)では、ID=IDsのスライド部分に対応するDIVタグの位置をステップ数だけ変更する。図8

(b)のSetSliderValue(IDs, 新しい値)は、ID=IDsのスライド部分に対応するDIVタグの位置を“新しい値”に対応する位置に変更する。図8(c)のGetSliderValue(IDs)は、ID=IDsのスライド部分に対応するDIVタグの位置から、スライドの値を判定し返す。図8(d)のSetPortion(IDs, 新しい値)はスライドType=Proportionalの場合のみ有効であり、ID=IDsに対応するインジケータの可動方向軸に対する表示範囲の大きさを変更する。図8(e)のGetPortion(IDs)はスライドType=Proportionalの場合のみ有効でありID=IDsに対応するインジケータの可動方向軸に対する表示範囲の大きさを返す。

【0048】図9にスライドオブジェクトおよびスライドオブジェクトに係るアクションを記述したMHEG-5形式の文書の例を示す。また、このMHEG-5形式の文書を変換した結果のHTML 4.0 set文書の例を図10~15に示す。このHTML 4.0 set文書は、実際にブラウザ上で表示できることを確認したものである。サンプル中にBitmapオブジェクトがあるが、BitmapのようなMHEG-5とHTML 4.0 setで1対1に対応するオブジェクトは属性値もほぼ1対1に対応するため、直接的に変換を行った。

【0049】図9のMHEG-5文書には、横に並ぶ3つのスライドが記述されている。左端のスライドのタイプはNormalであり、真ん中のスライドのタイプはThermometerであり、右端のスライドのタイプはProportionalであ

る。また、左端のスライドの下に配置されるボタンのビットマップも記述されている。Link100には、ビットマップにUserInputのイベントが発生したときに左端のスライドをステップ移動させることが記述されている。Link110には、真ん中のスライドにCursorEnterのイベントが発生したときに、真ん中のスライドの値を右端のスライドのインジケータのPortionに設定することが記述されている。Link120には、右端のスライドにCursorEnterのイベントが発生したときに、左端のスライドのインジケータのPortionの値を真ん中のスライドに設定することが記述されている。

【0050】図10~15のHTML 4.0 set文書では、StepSize、Orientation、SliderType、Pointは配列により管理されている。Step、SetSliderValue、GetSliderValue、SetPortion、GetPortionはJavaScriptにより記述されるメソッドであり、MHEG-5文書中のリンクイフェクトの各要素のメソッドに対応している。これらのメソッド中でも、スライドは配列により管理されている。

【0051】ID=ID7000o、ID=ID7000sのDIVタグは、MHEG-5文書中のSlider7000に対応するものであり、ID=ID7000oは、スライドの枠を表し、ID=ID7000sは、スライドのインジケータを表す。同様に、ID=ID7001o、ID=ID7001sのDIVタグは、MHEG-5文書中のSlider7001に対応するものであり、ID=ID7001oは、スライドの枠を表し、ID=ID7001sは、スライドのインジケータを表す。更に、ID=ID7002o、ID=ID7002sのDIVタグは、MHEG-5文書中のSlider7002に対応するものであり、ID=ID7002oは、スライドの枠を表し、ID=ID7002sは、スライドのインジケータを表す。

【0052】ID=ID7001sのDIVタグのOnMouseOverのイベントハンドラには、SetPortionとGetPortionより成るメソッドの組み合わせが関連づけられている。ID=ID7001sのDIVタグは、MHEG-5文書のLINK110のEventSourceに対応する。OnMouseOverは、MHEG-5文書のLINK110のイベントタイプであるCursorEnterに対応する。メソッドであるSetPortionとGetPortionは、MHEG-5文書のLINK110のLinkEffectに対応する。

【0053】ID=ID7002sのDIVタグのOnMouseOverのイベントハンドラには、SetSliderValueとGetPortionより成るメソッドの組み合わせが関連づけられている。ID=ID7002sのDIVタグは、MHEG-5文書のLINK120のEventSourceに対応する。OnMouseOverは、MHEG-5文書のLINK120のイベントタイプであるCursorEnterに対応する。メソッドであるSetSliderValueとGetPortionは、MHEG-5文書のLINK120のLinkEffectに対応する。

【0054】SRC="button.bmp"のIMGタグのOnClickのイベントハンドラには、メソッドであるStepが関連づけられている。SRC="button.bmp"のIMGタグはMHEG-5文書のLINK100のEventSourceに対応する。OnClickは、MHEG-5文書のLINK100のEventTypeに対応する。メソッドであるSt

epはMHEG-5文書のLINK100のLinkEffectに対応する。

【0055】なお、上記の図10～15の説明では、ID=7001sのDIVタグ節にOnMouseOverのイベントハンドラを、ID=7002sのDIVタグ節にOnMouseOverのイベントハンドラを、の節にOnClickのイベントハンドラを挿入したが、イベントハンドラの挿入位置はDIVタグ節、IMG節に限られるものではなく、OBJECT節その他のタグ節でもよい。これらのタグ節を基にブラウザによりHTML上のオブジェクトが生成される。

【0056】また、上記の実施形態では、スライダオブジェクトの形状を2つのDIVタグ節に変換したが、スライダオブジェクトのタイプが、ノーマルである場合には、スライダオブジェクトの形状をインジケータに対応したDIVタグ節、枠部分の片端からインジケータまでの部分に対応したDIVタグ節、及び枠部分の他端からインジケータまでの部分に対応したDIVタグ節に変換して、後2者のOnClickイベントハンドラにSTEPのメソッドを対応付けることもできる。

【0057】スライダオブジェクトに関しては、1つのスライダオブジェクトからHTML 4.0 set上の複数のオブジェクトを生成するとしたが、MHEG文書上のオブジェクトの種類に応じて、MHEG-5文書上のオブジェクトから1つのHTML 4.0 set上のオブジェクトを生成しても良い。

【0058】〔実施形態2〕次に、本発明の実施形態2について、図面を参照して説明する。

【0059】図16乃至21に、本発明の実施形態2による各MHEGオブジェクトの各プロパティの変換表を示す。

【0060】図16乃至21において、左端欄はMHEGオブジェクト、中欄はMHEGオブジェクトのプロパティ、右端欄は変換後のHTML 4.0 set上での表現形式を示す。

【0061】CursorShapeオブジェクトは、<OBJECT>タグによって囲まれる記述もしくは、CSSの属性に変換される。<OBJECT>タグによって囲まれる記述に変換される場合は、ObjectIdentifierは、IDに変換される。InitiallyActiveは、スタイル属性のvisibilityに変換される。ContentHookは、スクリプトのCLASSIDに変換される。OrigContentは、PARAMタグに変換される。PARAMタグ内のfilename等は、Cursorの形状を示すビットマップファイル名などである。Sharedは、変換の処理中にflag情報として持たれ、使用する場所で<OBJECT>タグによって囲まれる記述に変換される。

【0062】Paletteオブジェクトは、HTML 4.0 set上の特定のオブジェクトには変換できず、関連する個々のオブジェクトのスタイル属性であるstyle.colorに変換される。

【0063】Fontオブジェクトは、CursorShapeと同様

に変換される。

【0064】BooleanVariableオブジェクト、IntegerVariableオブジェクト、OctetStringVariableオブジェクト、ObjectRefVariableオブジェクト、ContentRefVariableオブジェクトは、一種の変数オブジェクトであり、そのObjectIdentifierは、スクリプトの変数に変換され、そのOriginalValueは、その変数の値に変換される。

【0065】TokenGroupオブジェクトのTokenGroupItemsにあるオブジェクトに対応したHTML 4.0 set文書上のオブジェクトには、連番IDが付され、そのMovementTableは、SetFocusを用いたスクリプトによるメソッドに変換され、そのメソッドには、TokenGroupオブジェクトの名称であるObjectIdentifierが付される。ListGroupについても同様である。

【0066】DynamicLineArtオブジェクトのOriginalPaletteRef, BorderedBoundingBox, OriginalLineWidth, OriginalLineStyle, OriginalRefLineColor, OriginalRefFillColorは、プラグインオブジェクトに変換される。PARAMに続くかっこ内で示されるのは、HTML文書とプラグインオブジェクトとの間の引数である。

【0067】図22乃至24に、本発明の実施形態2によるMHEGの各アクションの変換表を示す。アクションは、リンクオブジェクトのリンクイフェクトで使われる。

【0068】AddItemアクションに対応する連番IDはタグであり、AddItemは、スクリプトで記述される。

【0069】Deselectは、OnMouseClickやOnKeyDownなどのイベントハンドラに関連付けられたスクリプトに変換されるが、オブジェクトにより、イベントハンドラやスクリプトの種類が異なる。

【0070】DrawArcは、プラグインオブジェクトにより記述され、<OBJECT>で示すのはそのプラグインオブジェクトの種類、<PARAM>で示すのはプラグインオブジェクトと交換する情報である。

【0071】図25に、本発明の実施形態2によるMHEGの各イベントの変換表を示す。イベントは、リンクオブジェクトのイベントタイプに使用される。

【0072】上記の実施形態1、2では、JavaScriptをHTML文書に組み込んでいたが、HTML文書とは別個のファイルに書き込んで、HTML文書から呼び出すようにしても良い。

【0073】また、上記の実施形態1、2においては、スクリプト言語としてJavaScriptを使用した。VBScriptやPerlなどの他のスクリプト言語を使用することもできる。

【0074】なお、上記の実施形態による方法は、CPUに当該方法を実行させるためのプログラムを記録したCD-ROMなどの記録媒体からCPUが当該プログラムを読み込んで実行することにより実現することもでき

る。

【0075】また、上記の実施形態による方法は、図26に示すように、CPU203に当該方法を実行させるためのプログラム207を記録したCD-ROMなどのプログラム記録媒体206からCPU203が当該プログラム207を読み込んで実行することにより実現することもできる。この際、通常は、CPU203は、オペレーティングシステムにより、当該プログラム207を当該記録媒体206からメインメモリ204に一旦転送した後に、メインメモリ204から当該プログラムを読み込んで実行する。CPUとメインメモリ204を備えるコンピュータは、文書変換装置として機能し、MHEG-5文書201を入力し、これを装置内でHTML 4.0 set文書に変換した後、HTML 4.0 set文書205を出力する。

【0076】また、上記プログラムは、インタプリタ形式の命令を記述したテキスト形式のファイルと、このファイルを読み込んで実行命令に変換するインタプリタより生成されることもある。

【0077】

【発明の効果】以上説明したように、本発明によれば、MHEG文書からHTML文書への変換において、部品として変換することができないMHEGオブジェクトを変換することができる。

【図面の簡単な説明】

【図1】本発明の実施形態による文書変換方法における、オブジェクトを変換する方法を説明するための概念図である。

【図2】本発明の実施形態1で扱うスライダの形状を示す図である。

【図3】本発明の実施形態1によるスライダの属性の変換表である。

【図4】本発明の実施形態1で扱うスライダの種類を表す図である。

【図5】MHEG-5により記述されるスライダと、これを基に本発明の実施形態1による方法により変換したHTML 4.0 setで記述されるスライダの形状を示す図である。

【図6】本発明の実施形態1によるアクションの変換表である。

【図7】本発明の実施形態1による方法により生成されるJavaScriptにより記述される変数とスクリプトを示す図である。

【図8】本発明の実施形態1による方法により生成されるJavaScriptにより記述されるスクリプトを示す図である。

【図9】本発明の実施形態1による方法を適用するMHEG-5文書の例を示す図である。

【図10】図9に示すMHEG-5文書を本発明の実施形態1による方法で変換したHTML 4.0 set文書を示す図の第1部分である。

【図11】図9に示すMHEG-5文書を本発明の実施形態1

による方法で変換したHTML 4.0 set文書を示す図の第2部分である。

【図12】図9に示すMHEG-5文書を本発明の実施形態1による方法で変換したHTML 4.0 set文書を示す図の第3部分である。

【図13】図9に示すMHEG-5文書を本発明の実施形態1による方法で変換したHTML 4.0 set文書を示す図の第4部分である。

【図14】図9に示すMHEG-5文書を本発明の実施形態1による方法で変換したHTML 4.0 set文書を示す図の第5部分である。

【図15】図9に示すMHEG-5文書を本発明の実施形態1による方法で変換したHTML 4.0 set文書を示す図の第6部分である。

【図16】本発明の実施形態2による各MHEGオブジェクトの各プロパティの変換表の第1部分である。

【図17】本発明の実施形態2による各MHEGオブジェクトの各プロパティの変換表の第2部分である。

【図18】本発明の実施形態2による各MHEGオブジェクトの各プロパティの変換表の第3部分である。

【図19】本発明の実施形態2による各MHEGオブジェクトの各プロパティの変換表の第4部分である。

【図20】本発明の実施形態2による各MHEGオブジェクトの各プロパティの変換表の第5部分である。

【図21】本発明の実施形態2による各MHEGオブジェクトの各プロパティの変換表の第6部分である。

【図22】本発明の実施形態2によるMHEGの各アクションの変換表の第1部分である。

【図23】本発明の実施形態2によるMHEGの各アクションの変換表の第2部分である。

【図24】本発明の実施形態2によるMHEGの各アクションの変換表の第3部分である。

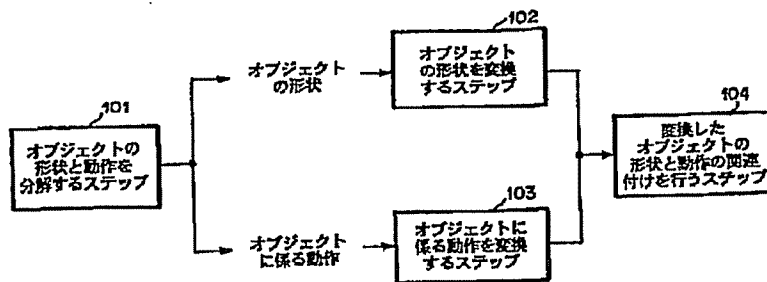
【図25】本発明の実施形態2によるMHEGの各イベントの変換表である。

【図26】本発明の実施形態による方法を実現するための装置を示すブロック図である。

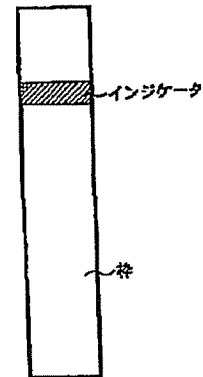
【符号の説明】

- 101 オブジェクトの形状と動作を分解するステップ
- 102 オブジェクトの形状を変換するステップ
- 103 オブジェクトに係る動作を変換するステップ
- 104 変換したオブジェクトの形状と動作の関連付けを行うステップ
- 201 MHEG-5文書
- 202 文書変換装置
- 203 CPU
- 204 メインメモリ
- 205 HTML 4.0 set文書
- 206 プログラム記録媒体
- 207 プログラム

【図1】



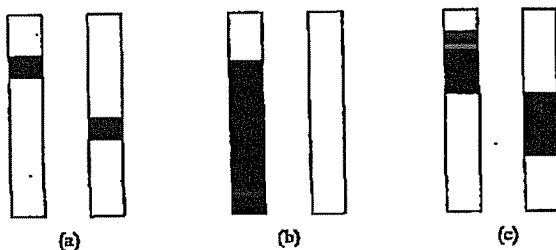
【図2】



【図3】

MHEG-5		HTML 4.0 set	
属性	種別	表現	備考
ObjectIdentifier	M	ID	オブジェクトID(後、スライド部分に適用)
InitiallyActive	O(True)	style.visibility	可視性の指定
Shared	O(False)	※	複数シーンページで使用する場合にTrue
OriginalBoxSize	M	style.width, style.height	外枠部分の大きさ
OriginalPosition	O(0, 0)	style.left, style.top	外枠部分の位置
OriginalPaletteRef	O(NoPalette)	※	パレットを使用する場合は使用されている
EngineResp	O(True)	—	—
HighlightRefColor	O(applicationクラス依存)	style.borderColor	ハイライトの色
Orientation	M	※	スライドの移動方向
InitialValue	O(MinValue)	style.width, style.height	スライド部分の位置
MinValue	O(1)	style.width, style.height	外枠部分の大きさ
MaxValue	M	style.width, style.height	外枠部分の大きさ
InitialPortion	O(None)	style.width, style.height	スライド部分の大きさ(proportional)
StepSize	O(1)	※	スライドの移動間隔
SliderStyle	O(normal)	style.width, style.height	スライド部分の形、大きさ
SliderRefColor	O(applicationクラス依存)	style.color	スライドの色

【図4】



【図6】

MHEG-5		HTML 4.0 set	
アクション	記述	JavaScript関数	備考
Step	:Step(234, 2)	Setp(ID, ステップ数)	ステップ数だけIDで示されたスライドを変化
SetSliderValue	:SetSliderValue(234, 20)	SetVal(ID, 新しい値)	新しい値で示された位置にIDで示されたスライドを変化
GetSliderValue	:GetSliderValue(234, 1)	返り値=GetVal(ID)	IDで示されたスライドの位置を返す
SetPortion	:SetPortion(234, 5)	SetPor(ID, 新しい値)	インジケータの大きさを返す(Proportionalの場合のみ)
GetPortion	:GetPortion(234, 1)	返り値=GetPor(ID)	インジケータの大きさを返す(Proportionalの場合のみ)

【図5】

```

{:Slider 7901
  :InitiallyActive true
  :OriginalBoxSize (10 110)
  :OriginalPosition (20 20)
  :OriginalPaletteRef 1
  :EngineResp true
  :HighlightColour #FFFFFF
  :Orientation left
  :InitialValue 50
  :MinValue 0
  :MaxValue 100
  :InitialPortion 20
  :StepSize 5
  :SliderStyle Proportional
  :SliderRefColour #FF0000
}

```

(a)

```

<DIV
  ID=7901o
  style = "visibility: true;
  width: 10; height: 100;
  left: 10; top: 10;
  border: #FFFFFF;"
/>
<DIV
  ID=7901s
  style = "visibility: true;
  width: 10; height: 20;
  left: 20; top: 20+50-0.5*20;
  border: #FFFFFF;"
/>

```

(b)

【図15】

```

ID=ID7002s
style="visibility: visible;
position: absolute;
width: 20; height: 40;
left: 210; right: 240; top: 50; bottom: 70;
background: #0000FF; z-index: 1"
onmouseover
="SetSliderValue(7001, ID7001s, ID7001o, GetPortion(7002, ID7002s))"> </DIV>
<IMG
SRC="button.bmp"
ID=ID8000
style="visibility: visible;
position: absolute;
width: 40; height: 20;
left: 0; top: 230;"
OnClick="Step(7000, ID7000s, 2)" />
</BODY>
</HTML>

```

【図7】

```

function Initialize()
{
    var StepSize = StepSize
    var Orientation = Orientation
    var SliderType = SliderType
}

```

(X)

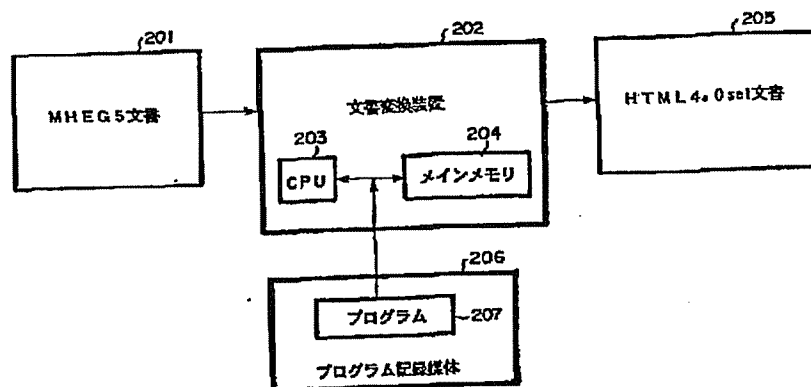
```

function Step(IDs, ステップ数) {
    var stepnum = ステップ数
    switch(Orientation){
        case left:
            IDs.style.left = IDs.style.left - StepSize * stepnum;
            break;
        case right:
            IDs.style.left = IDs.style.left + StepSize * stepnum;
            break;
        case up:
            IDs.style.top = IDs.style.top - StepSize * stepnum;
            break;
        case down:
            IDs.style.top = IDs.style.top + StepSize * stepnum;
            break;
    }
}

```

(a)

【図26】



【図8】

```

function SetSliderValue(IDs, 新しい値){
  var newval = 新しい値
  switch(Orientation){
    case left:
      IDs.style.left = IDo.style.right - newval
      break
    case right:
      IDs.style.left = IDo.style.left + newval
      break
    case up:
      IDs.top = IDo.style.bottom - newval
      break
    case down:
      IDs.top = IDo.style.top + newval
      break
  }
}

```

(b)

```

function GetSliderValue(IDs){
  switch(Orientation){
    case left:
      return IDo.right - IDs.left
      break
    case right:
      return IDs.left - IDo.left
      break
    case up:
      return IDs.bottom - IDo.top
      break
    case down:
      return IDs.top - IDo.top
      break
  }
}

```

(c)

```

function SetPortion(IDs, 新しい値){
  var PortSize = 新しい値
  switch(Orientation){
    case left:
    case right:
      var now = IDs.right - IDs.left
      var d = now - PortSize/2
      IDs.left = IDs.left - d
      IDs.right = IDs.right + d
      break
    case up:
    case down:
      var now = IDs.right - IDs.left
      var d = now - PortSize/2
      IDs.left = IDs.left - d
      IDs.right = IDs.right + d
      break
  }
}

```

(d)

```

function GetPortion(IDs){
  switch(Orientation){
    case left:
    case right:
      return IDs.left - IDs.right
      break
    case up:
    case down:
      return IDs.bottom - IDs.top
      break
  }
}

```

(c)

【図9】

```

{:Scene ("~/Scene1" 0 )
:StdVersion 1
:Items (
  {Slider 7000
    :InitiallyActive true
    :OriginalBoxSize (10 100)
    :OriginalPosition (10 10)
    :OriginalPaletteRef 1
    :EngineResp true
    :HighlightColour #FFFFFF
    :Orientation up
    :InitialValue 50
    :MinValue 0
    :MaxValue 100
    :StepSize 5
    :SliderStyle Normal
    :SliderRefColour #FF0000
  }
  {Slider 7001
    :InitiallyActive true
    :OriginalBoxSize (10 100)
    :OriginalPosition (110 10)
    :OriginalPaletteRef 1
    :EngineResp true
    :HighlightColour #FFFFFF
    :Orientation up
    :InitialValue 50
    :MinValue 0
    :MaxValue 100
    :StepSize 5
    :SliderStyle Thermometer
    :SliderRefColour #00FF00
  }
  {Slider 7002
    :InitiallyActive true
    :OriginalBoxSize (10 100)
    :OriginalPosition (210 10)
    :OriginalPaletteRef 1
    :EngineResp true
    :HighlightColour #FFFFFF
    :Orientation up
    :InitialValue 50
    :InitialPortion 20
    :MinValue 0
    :MaxValue 100
    :StepSize 5
    :SliderStyle Proportional
    :SliderRefColour #0000FF
  }
  {Bitmap 8000
    :InitiallyActive true
    :Chook 4
    :OrigContent :ContentRef ("~/button.png")
    :OrigBoxSize 50 10
    :OrigPosition 10 130
    :Tiling false }
  {Link 100
    :EventSource 8000
    :EventType UserInput
    :EventData 1
    :LinkEffect (
      :Step (7000 2) )}
  {Link 110
    :EventSource 7001
    :EventType CursorEnter
    :LinkEffect (
      :GetSliderValue(7001 25)
      :SetPortion (7002 :IndirectRef 25) )}
  {Link 120
    :EventSource 7002
    :EventType CursorEnter
    :LinkEffect (
      :GetPortion(7002 26)
      :SetSliderValue (7001 :IndirectRef 26) )}
  {IntegerVar 25
    :OrigValue 0
  }
  {IntegerVar 26
    :OrigValue 0
  }
  :InputEventReg 1
  :SceneCS 640 480
  :MovingCursor true }
}

```

【図10】

```

<HTML>
<HEAD>
<SCRIPT>
function InitArrayQ {
    this.length = InitArray.arguments.length
    for(var i = 0; i < this.length; i++)
        this[i+1] = InitArray.arguments[i]
}

function InitializeQ {
    StepSize = new InitArray(5, 5)
    Orientation = new InitArray("up", "up", "up")
    SliderType = new InitArray("Normal", "Thermometer", "Proportional")
    Point = new InitArray(7000, 7001, 7002)
}

function Step(ID, IDs, stepnum) {
    for(i = 1; i <= 3; i++) {
        if(Point[i] == ID) {
            num = i
            break
        }
    }

    if(SliderType[num] == "Thermometer") {
        switch(Orientation[num]) {
            case "left":
                IDs.style.width = StepSize[num]*stepnum + IDs.style.width, toStringQ.substring(0, IDs.style.width.length-2)
                break
            case "right":
                IDs.style.width = StepSize[num]*stepnum + IDs.style.width, toStringQ.substring(0, IDs.style.height.length-2) = (-StepSize[num]*stepnum)
                break
            case "up":
                IDs.style.height = IDs.style.height, toStringQ.substring(0, IDs.style.height.length-2) = (-StepSize[num]*stepnum)
                break
            case "down":
                break
        }
    }

    switch(Orientation[num]) {
        case "left":
            tmp = IDs.style.left.toStringQ.substring(0, IDs.style.left.length-2)
            IDs.style.left=tmp-StepSize[num]*stepnum
    }
}

```

【図11】

```

        break
    case "right":
        tmp = IDs.style.left.toString().substring(0, IDs.style.left.length-2)
        IDs.style.left=tmp-(-StepSize[num]*stepnum)
        break
    case "up":
        tmp = IDs.style.top.toString().substring(0, IDs.style.top.length-2)
        IDs.style.top = tmp-StepSize[num]*stepnum
        break
    case "down":
        tmp = IDs.style.top.toString().substring(0, IDs.style.top.length-2)
        IDs.style.top = tmp-(-StepSize[num]*stepnum)
        break
    }
}

function SetSliderValue(ID, IDs, IDo, newval) {
    for(i = 1; i <= 3; i++){
        if(Point[i] == ID){
            num = i
            break
        }
    }

    switch(Orientation[num]){
    case "left":
        IDs.style.left = IDo.style.right.toString().substring(0, IDo.style.right.length-2) - newval
        IDs.style.widtht = newval
        break
    case "right":
        IDs.style.right = IDo.style.left.toString().substring(0, IDo.style.left.length-2) + newval
        IDs.style.widtht = newval
        break
    case "up":
        IDs.style.top = IDo.style.bottom.toString() - newval
        IDs.style.height = newval
        break
    case "down":
        IDs.style.bottom = IDo.style.top.toString() + newval
    }
}

```


【図12】

```

        IDs.style.height = newval
        break;
    }
}

function GetSliderValue(ID, IDs, IDo) {
    for(i = 1; i <= 3; i++) {
        if(Point[i] == ID) {
            num = i
            break
        }
    }

    switch(Orientation[num]) {
        case "left":
            orgt = IDo.style.right.toString().substring(0, IDo.style.right.length-2)
            slft = IDs.style.left.toString().substring(0, IDs.style.left.length-2)
            return (orgt - slft)
        case "right":
            srgt = IDs.style.rgt.toString().substring(0, IDs.style.rgt.length-2)
            olft = IDo.style.left.toString().substring(0, IDo.style.left.length-2)
            return (srgt - olft)
        case "up":
            obtm = IDs.style.bottom.toString()
            stop = IDs.style.top.toString().substring(0, IDs.style.top.length-2)
            return (obtm - stop)
        case "down":
            sbtm = IDs.style.bottom.toString().substring(0, IDs.style.bottom.length-2)
            otop = IDo.style.top.toString().substring(0, IDo.style.top.length-2)
            return (sbtm - otop)
    }
}

function SetPortion(ID, IDs, IDo, PortSize) {
    for(i = 1; i <= 3; i++) {
        if(Point[i] == ID) {
            num = i
            break
        }
    }
}

```

【図13】

```

if(SliderType[num] == "Proportional"){
switch(Orientation[num]){
case "left":
now = IDs.style.width.toString().substr(0, IDs.style.width.length-2)
d = now - PortSize
cc = d/2
IDs.style.left = IDs.style.left.toString().substr(0, IDs.style.left.length-2) - dd
IDs.style.width = IDs.style.width.toString().substr(0, IDs.style.width.length-2) - d
break
case "up":
case "down":
now = IDs.style.height.toString().substr(0, IDs.style.height.length-2)
d = now - PortSize
dd = d/2
IDs.style.top = IDs.style.top.toString().substr(0, IDs.style.top.length-2) - dd
IDs.style.height = IDs.style.height.toString().substr(0, IDs.style.height.length-2) - d
break
}
}
}

function GetPortion(ID, IDs){
for(i = 1; i <= 3; i++){
if(Point[i] == ID){
num = i
break
}
}
}

if(SliderType[num] == "Proportional"){
switch(Orientation[num]){
case "left":
case "right":
return IDs.style.width.toString().substr(0, IDs.style.width.length-2)
case "up":
case "down":
return IDs.style.height.toString().substr(0, IDs.style.height.length-2)
}
}
}

```

【図14】

```
</SCRIPT>
</HEAD>

<BODY onLoad="Initialize()">
<DIV
  ID=ID7000o
  style="visibility: visible;
  position: absolute;
  width: 20; height: 200;
  left: 10; right: 40; top: 10; bottom: 210;
  border: solid 1px #000000;"> </DIV>
<DIV
  ID=ID7000s
  style="visibility: visible;
  position: absolute;
  width: 20; height: 5;
  left: 10; right: 40; top: 60; bottom: 61;
  background: #FF0000;"> </DIV>
<DIV
  ID=ID7001o
  style="visibility: visible;
  position: absolute;
  width: 20; height: 200;
  left: 110; right: 140; top: 10; bottom: 210;
  border: solid 1px #000000;"> </DIV>
<DIV
  ID=ID7001s
  style="visibility: visible;
  position: absolute;
  width: 20; height: 50;
  left: 110; right: 140; top: 160; bottom: 210;
  background: #00FF00;
  onmouseover="SetPortion(7002, ID7002s, ID7002o, GetSliderValue(7001, ID7001s, ID7001o))"> </DIV>
<DIV
  ID=ID7002o
  style="visibility: visible;
  position: absolute;
  width: 20; height: 200;
  left: 210; right: 240; top: 10; bottom: 210;
  border: solid 1px #000000;"> </DIV>
<DIV
```

【図16】

AREG-5	HTML4.0src1
CursorShape	<OBJECT>
ObjectIdentifier	ID
InitiallyActive	style, visibility
ContentHook	CLASSID
OrigContent	PARAM(filename等)
Shared	1. flag情報として持つ 2. 使用する場所で<OBJECT>に変換
Palette	1. 参照しているパレットをrgbパレットに変換
ObjectIdentifier	2. 使用する場所でstyle.color等に変換
InitiallyActive	
ContentHook	
OrigContent	
Shared	
Font	<OBJECT>
ObjectIdentifier	ID
InitiallyActive	style, visibility
ContentHook	CLASSID
OrigContent	PARAM(filename等)
Shared	1. flag情報として持つ 2. 使用する場所で<OBJECT>に変換
BooleanVariable	
ObjectIdentifier	変数名
InitiallyActive	- (最初に定義)
Shared	-
OriginalValue	変数値
IntegerVariable	
ObjectIdentifier	変数名
InitiallyActive	- (最初に定義)
Shared	-
OriginalValue	変数値
OctetStringVariable	
ObjectIdentifier	変数名
InitiallyActive	- (最初に定義)
Shared	-
OriginalValue	変数値
ObjectRefVariable	
ObjectIdentifier	変数名1, 変数名2
InitiallyActive	- (最初に定義)
Shared	-
OriginalValue	変数値1, 変数値2

【図17】

ContentRefVariable	ObjectIdentifier	変数名1, 変数名2
	InitiallyActive	- (最初に定義)
	Shared	-
	OriginalValue	変数値1, 変数値2
TokenGroup	ObjectIdentifier	関数名
	InitiallyActive	- (最初に定義)
	ContentHook	-
	OriginalContent	-
	Shared	-
	MovementTable	JavaScript
	TokenGroupItems	連番ID + style.position
	NoTokenActionSlots	-
ListGroup	ObjectIdentifier	関数名
	InitiallyActive	- (最初に定義)
	ContentHook	-
	OriginalContent	-
	Shared	-
	MovementTable	JavaScript
	TokenGroupItems	連番ID
	NoTokenActionSlots	-
	Positions	style.position
	WrapAround	style.position
	MultipleSelection	JavaScript
Bitmap		<OBJECT>
	ObjectIdentifier	ID
	InitiallyActive	style.visibility
	ContentHook	CLASSID
	OrigContent	PARAM(filename等)
	Shared	1. flag情報として持つ 2. 使用する場所で<OBJECT>に変換
	OriginalBoxSize	style.width, style.height
	OriginalPosition	style.left, style.top
	OriginalPaletteRef	-
	Tiling	複数の<OBJECT>に変換
	OrigTransparency	-

【図18】

LineArt	<OBJECT>	
	ObjectIdentifier	ID
	InitiallyActive	style.visibility
	ContentHook	CLASSID
	OrigContent	PARAM(filename等)
	Shared	1. flag情報として持つ 2. 使用する場所で<OBJECT>に変換
	OriginalBoxSize	style.width, style.height
	OriginalPosition	style.left, style.top
	OriginalPaletteRef	PARAM(palette等)
	BorderedBoundingBox	PARAM(BBBox等)
	OriginalLineWidth	PARAM(width等)
	OriginalLineStyle	PARAM(style等)
	OriginalRefLineColor	PARAM(Lcolor等)
	OriginalRefFillColor	PARAM(Fcolor等)
Rectangle	<DIV>	
	ObjectIdentifier	ID
	InitiallyActive	style.visibility
	Shared	1. flag情報として持つ 2. 使用する場所で<TABLE>に変換
	OriginalBoxSize	style.width, style.height
	OriginalPosition	style.left, style.top
	OriginalPaletteRef	参照するrgbパレット
	OriginalLineWidth	style.border
	OriginalLineStyle	style.border
	OriginalRefLineColor	style.bordercolor ← rgbパレット
	OriginalRefFillColor	style.background ← rgbパレット
DynamicLineArt	<OBJECT>	
	ObjectIdentifier	ID
	InitiallyActive	style.visibility
	Shared	CLASSID
	OriginalBoxSize	style.width, style.height
	OriginalPosition	style.left, style.top
	OriginalPaletteRef	PARAM(palette等)
	BorderedBoundingBox	PARAM(BBBox等)
	OriginalLineWidth	PARAM(width等)
	OriginalLineStyle	PARAM(style等)
	OriginalRefLineColor	PARAM(Lcolor等)
	OriginalRefFillColor	PARAM(Fcolor等)

【図19】

Text		<DIV>
	ObjectIdentifier	ID
	InitiallyActive	style.visibility
	ContentHook	-
	OriginalContent	テキスト文書
	Shared	1. flag情報として持つ 2. 使用する場所で<DIV>+文書に変換
	OriginalBoxSize	style.width, style.height
	OriginalPosition	style.left, style.top
	OriginalPaletteRef	参照するrgbパレット
	OriginalFont	style.fontfamily
	FontAttributes	style.fontstyle, style.fontsize
	TextColour	style.color ← rgbパレット
	BackGroundColour	style.background ← rgbパレット
	CharacterSet	-
	HorizontalJustification	style.text-align
	VerticalJustification	style.vertical-align
	LineOrientation	-
	StartCorner	-
	TextWrapping	-
Stream		<OBJECT>
	ObjectIdentifier	ID
	InitiallyActive	style.visibility
	ContentHook	CLASSID
	OriginalContent	PARAM(filename等)
	Shared	1. flag情報として持つ 2. 使用する場所で<OBJECT>に変換
	Multiplex	-
	Storage	-
	Looping	PARAM(looping等)
Audio		<OBJECT>
	ObjectIdentifier	ID
	InitiallyActive	style.visibility
	ComponentTag	CLASSID
	OriginalVolume	PARAM(volume等)
Video		<OBJECT>
	ObjectIdentifier	ID
	InitiallyActive	style.visibility
	OriginalBoxSize	style.width, style.height
	OriginalPosition	style.top, style.left
	ComponentTag	CLASSID
	Termination	PARAM(termination等)

【図20】

RTGraphics		<OBJECT>
	ObjectIdentifier	ID
	InitiallyActive	style.visibility
	OriginalBoxSize	style.width, style.height
	OriginalPosition	style.top, style.left
	ComponentTag	CLASSID
	Termination	PARAM(termination等)
EntryField		<FORM>
	ObjectIdentifier	ID
	InitiallyActive	style.visibility
	ContentHook	-
	OrigContent	VALUE
	Shared	1. flag情報として持つ 2. 使用する場所で<FORM>に変換
	OriginalBoxSize	style.width, style.height
	OriginalPosition	style.top, style.left
	OriginalPaletteRef	参照するrgbパレット
	OriginalFont	style.fontfamily
	FontAttributes	style.fontstyle, style.fontsize
	TextColour	style.color ← rgbパレット
	BackGroundColour	style.background ← rgbパレット
	CharacterSet	-
	HorizontalJustification	style.text-align
	VerticalJustification	style.vertical-align
	LineOrientation	-
	StartCorner	-
	TextWrapping	-
	EngineResp	-
	HighlightRefColour	style.bordercolor
	CharList	<OPTION>
	ObscuredInput	<INPUT TYPE='password'
	MaxLength	MAXLENGTH
Hotspot		<DIV>
	ObjectIdentifier	ID
	InitiallyActive	style.visibility
	Shared	1. flag情報として持つ 2. 使用する場所で<FORM>に変換
	OriginalBoxSize	style.width, style.height
	OriginalPosition	style.top, style.left
	OriginalPaletteRef	参照するrgbパレット
	EngineResp	-
	HighlightRefColour	style.bordercolor ← rgbパレット
	ButtonRefColour	style.color ← rgbパレット

【図21】

PushButton		<FORM> type='button'
	ObjectIdentifier	ID
	InitiallyActive	-
	Shared	1. flag情報として持つ 2. 使用する場所で<FORM>に変換
	OriginalBoxSize	style.width, style.height
	OriginalPosition	style.top, style.left
	OriginalPaletteRef	参照するrgbパレット
	EngineResp	-
	HighlightRefColour	style.bordercolor ← rgbパレット
	ButtonRefColour	style.color ← rgbパレット
	OriginalLabel	name
	CharacterSet	-
SwitchButton		<FORM> type='radio'
	ObjectIdentifier	ID
	InitiallyActive	-
	Shared	1. flag情報として持つ 2. 使用する場所で<FORM>に変換
	OriginalBoxSize	style.width, style.height
	OriginalPosition	style.top, style.left
	OriginalPaletteRef	参照するrgbパレット
	EngineResp	-
	HighlightRefColour	style.bordercolor ← rgbパレット
	ButtonRefColour	style.color ← rgbパレット
	OriginalLabel	name
	CharacterSet	-
	ButtonStyle	-

【図22】

WMEG-S	HTMLA.0
Action	Activate
Add	"t"
AddItem	連番ID + JavaScript
Append	var
BringToFront	z-index
Call	-
CallActionSlot	pageX, pageY + ID + JavaScript
Clear	<OBJECT><PARAM>
Clone	-
CloseConnection	-
Deactivate	-
DeleteItem	連番ID + JavaScript
Deselect	OnMouseClicked, OnKeyDown等
DeselectItem	連番ID + JavaScript
Divide	"/"
DrawArc	<OBJECT><PARAM>
DrawLine	<OBJECT><PARAM>
DrawOval	<OBJECT><PARAM>
DrawPolygon	<OBJECT><PARAM>
DrawPolyline	<OBJECT><PARAM>
DrawRectangle	<OBJECT><PARAM>
DrawSector	<OBJECT><PARAM>
Fork	-
GetAvailabilityStatus	-
GetBoxSize	style, width, style, height
GetCellItem	連番ID + JavaScript
GetCursorPosition	pageX, pageY
GetEngineSupport	-
GetEntryPoint	-
GetFillColor	<OBJECT><PARAM>, style, background
GetFirstItem	連番ID + JavaScript
GetHighlightStatus	-
GetInteractionStatus	-
GetItemStatus	-
GetLabel	<FORM> type='button'
GetLastAnchorFired	-
GetLineColor	<OBJECT><PARAM>, style, bordercolor
GetLineStyle	<OBJECT><PARAM>, style, borderstyle
GetLineWidth	<OBJECT><PARAM>, style, border
GetListItem	連番ID + JavaScript

【図23】

GetListSize	var
GetOverrideMode	-
GetPortion	style, left, style, top, style, right, style, bottom
GetPosition	style, top, style, left
GetRunningStatus	JavaScript + visibility
GetSelectionStatus	JavaScript + visibility
GetSliderValue	style, left, style, top, style, right, style, bottom
GetTextContent	ID, var
GetTextData	ID, var
GetTokenPosition	pageX, pageY + ID
GetVariable	var
GetVolume	<OBJECT><PARAM>
Launch	location
LockScreen	-
Modulo	JavaScript
Move	ID + JavaScript + style, top等
MoveTo	ID + JavaScript + style, top等
Multiply	"x"
OpenConnection	-
Preload	-
PutBefore	style, z-index
PutBehind	style, z-index
Quit	-
ReadPersistent	-
Run	style, visibility
ScaleBitmap	style, width, style, height
ScaleVideo	style, width, style, height
ScrollItems	連番ID + JavaScript
Select	OnMouseClicked, OnKeyDown等
SelectItem	連番ID + JavaScript
SendEvent	-
SendToBack	style, z-index
SetBoxSize	style, width, style, height
SetCachePriority	-
SetCounterEndPosition	<OBJECT><PARAM>
SetCounterPosition	<OBJECT><PARAM>
SetCounterTrigger	<OBJECT><PARAM>
SetCursorPosition	-
SetCursorShape	style, cursor
SetData	-

【図24】

SetEntryPoint	-
SetFillColor	<OBJECT><PARAM>, style.background
SetFirstItem	連番ID + JavaScript
SetFontRef	style.font, style.font-family
SetHighlightStatus	-
SetInteractionStatus	-
SetLabel	<FORM> type='button'
SetLineColor	<OBJECT><PARAM>, style.borderColor
SetLineStyle	<OBJECT><PARAM>, style.borderstyle
SetLineWidth	<OBJECT><PARAM>, style.border
SetOverwriteNode	-
SetPaletteRef	-
SetPortion	style.left, style.top, style.right, style.bottom
SetPosition	style.top, style.left
SetSliderValue	style.left, style.top, style.right, style.bottom
SetSpeed	<OBJECT><PARAM>
SetTimer	SetTimeout
SetTransparency	-
SetVariable	var
SetVolume	var
Spawn	-
Step	style.left, style.top, style.right, style.bottom
Stop	style.visibility
StorePersistent	-
Subtract	"-"
TestVariable	var
Toggle	連番ID + JavaScript
ToggleItem	連番ID + JavaScript
TransitionTo	location
Unload	-
UnlockScreen	-

【図25】

Events	IsAvailable	style.visibility: true
	ContentAvailable	style.visibility: true
	IsDeleted	-
	IsRunning	style.visibility: true
	IsStopped	style.visibility: false
	UserInput	OnMouseClick, OnMouseKeyDown等
	AnchorFired	-
	TimerFired	SetTimeout
	AsynchStopped	-
	InteractionCompleted	-
	TokenMovedFrom	OnMouseOver, OnClick等
	TokenMovedTo	OnMouseOver, OnClick等
	StreamEvent	Plug-inによる
	StreamPlaying	Plug-inによる
	StreamStopped	Plug-inによる
	CounterTrigger	Plug-inによる
	HighlightOn	style.border
	HighlightOff	style.border
	CursorEnter	OnMouseOver
	CursorLeave	OnMouseOut
	IsSelected	OnClick, OnkeyDown等
	IsDeselected	OnClick, OnkeyDown等
	TestEvent	OnClick, OnkeyDown等
	FirstItemPresented	style.visibility: true
	LastItemPresented	style.visibility: true
	HeadItems	style.visibility: true
	TailItems	style.visibility: true
	ItemSelected	OnClick, OnkeyDown等
	ItemDeselected	OnClick, OnkeyDown等
	EntryFieldFull	-